

PLATAFORMA EDUCATIVA AMB NANO JETBOTS. PLANIFICACIÓ

Víctor Noguera López

Resum—A una plataforma ja creada l'any passat, destinada a que la facin servir alumnes de 4t de la ESO i els seus docents, es volen afegir mòduls. Es vol afegir un mòdul que recorri un laberint mitjançant una xarxa neuronal AlexNET que reconeixerà els diferents camins, i un algorisme de backtracking que recorrerà tot el laberint, per tant, els i les alumnes aprendran com funcionen tant la xarxa com l'algorisme de backtracking, així com a organitzar les dades. La plataforma està integrada en un jetbot i fa servir Jupyter Notebook, així que la programació que veuran i aprendran els i les alumnes serà Python. També s'afegirà un mòdul que reconeixerà senyals, forats i objectes, però aquest es realitzarà en el TFG de la Raquel Navarro, també d'enguany a la mateixa convocatòria. Finalment, s'afegirà un mòdul que unirà tots dos mòduls amb l'objectiu de trobar els objectes dins del laberint i cadascun donarà una sèrie de punts.

Paraules clau—Backtracking, Jetbot, AlexNET, xarxa neuronal, Jupyter Notebook, Python.

Abstract—On a platform already created last year, intended to be used by ESO 4th students and their teachers, some modules are to be added. I want to add a module that runs a labyrinth through an AlexNET neural network that will recognize the different path, and a backtracking algorithm that will follow the labyrinth path, so learners will learn how both the network and the backtracking algorithm work, as well as organizing the data. The platform is integrated into a jetbot and uses Jupyter Notebook, so the programming that will see and learn the students will be Python. A module that recognizes signals, holes and objects will also be added, but this will be done in the Raquel Navarro's TFG, also this year. Finally, a module will be added that will join both modules with the goal of finding objects inside the labyrinth and each one will give a number of points.

Index Terms— Backtracking, Jetbot, AlexNET, Neural Network, Jupyter Notebook, Python.

1 INTRODUCCIÓ I OBJECTIUS

ACTUALMENT la intel·ligència artificial és un camp que està en creixement i en un futur molt proper es necessitaran persones que siguin capaces de desenvolupar-les i d'entendre-les. Però actualment hi ha una manca de material adaptat i accessible a les edats on es comença a decidir el seu propi futur, així com una manca de coneixement sobre el tema. Per això es vol apropar la intel·ligència artificial a les noves generacions mitjançant una plataforma ja existent ^[1] afegint-hi uns mòduls més. D'aquesta manera volem aportar el nostre granet de sorra.

L'objectiu d'aquest TFG és el de treballar en una plataforma educativa amb nano Jetbots per poder treballar la intel·ligència artificial amb alumnes de tercer i quart de secundària. S'han generat diferents unitats o mòduls didàctics per tal que el professorat i l'alumnat de secundària puguin treballar la matèria de forma més o menys autònoma amb un conjunt de nano jetbots.

2 ESTAT DE L'ART

S'ha trobat que ja hi ha plataformes que apropen la robò-

tica als alumnes de secundària, tot i que no inclouen cap intel·ligència artificial o està molt limitada pel hardware, ja que no compten amb una targeta gràfica.

Alguns exemples serien:

1. Lego Education: Tenen els robots EV3 i Spike Prime ^[2]. Es fan servir peces Lego per construir els reptes proposats, fent que surtin robots molt diversos. A més, la seva interfície està dissenyada per als alumnes des de 10 anys d'edat, amb programació per blocs.



Figura 1. Spike Prime

- E-mail de contacte: victor.noguera@e-campus.uab.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Aura Hernández Sabaté (CVC)
- Curs 2020/21

2. mBots. Al contrari que l'anterior, aquest no compta amb peces Lego ni es pot modificar la construcció. Sí que té programació per blocs i està dissenyat per alumnes des de 10 anys també.



Figura 2. mBot

Ambdós exemples compten amb la possibilitat de programar en Python, però no tenen la capacitat de fer servir xarxes neuronals de forma eficient degut a la manca de targeta gràfica.

3 OBJECTIUS

S'han observat algunes de les mecàniques que ofereixen les altres plataformes i es volen millorar amb la realització del present TFG. Aquestes mecàniques fan servir diversos sensors, com el de color i/o ultrasònic, i no tenen un sistema d'emmagatzematge de dades prou gran. En aquest treball es fa servir únicament una càmera amb una xarxa neuronal.

S'ha separat la feina a fer dins de la plataforma en els següents mòduls:

- **Reconeixedor d'objectes i forats (mòdul 4):** l'objectiu d'aquest mòdul és poder fer que el nano Jetbot sigui capaç de reconèixer senyals, objectes i/o forats.
- **Resolució d'un laberint (mòdul 5):** l'objectiu d'aquest mòdul és poder fer que el nano Jetbot sigui capaç de reconèixer línies i a partir d'això poder arribar a resoldre el laberint. El robot va guardant el camí fet i retornarà a una intersecció quan arribi a un final de camí.
- **Unió dels 2 mòduls (mòdul 6):** després d'haver treballat en els 2 mòduls, els alumnes tindran l'oportunitat de combinar i fer servir les troballes dels mòduls anteriors.

Som dues persones treballant en aquesta plataforma, per tant s'han dividit les tasques per tal de poder assolir tots els objectius. Del reconeixedor d'objectes i forats s'ha encarregat la meua companya, Raquel Navarro, i de la resolució d'un laberint me n'he encarregat jo. Finalment, s'ha fet un tercer mòdul que consisteix en la unió dels dos anteriors.

Com ja s'ha dit anteriorment, cadascú té assignada una part del projecte a realitzar, tot i que l'altra persona ha estat de suport d'aquesta part.

Aprofitant que els mòduls són independents, cadascú ha pogut programar en Python i provar la seva part amb uns nano Jetbots i posar en comú una vegada s'ha tingut

una versió estable. Un cop finalitzat i revisat cada mòdul, s'han inclòs als altres nano Jetbots.

Com a requeriments es necessita:

- Per a les escoles, els nano Jetbots i d'un dispositiu que compta d'un navegador web, tot i que es recomana un ordinador o una tauleta amb teclat.
- Per a nosaltres, com a mínim un nano Jetbot i un ordinador amb un navegador web per programar i redactar la plataforma.

4 PLANIFICACIÓ

S'ha separat el projecte en tres mòduls, que han durat: deu setmanes el mòdul 4, dotze setmanes el mòdul 5 i tres setmanes el darrer mòdul (desenvolupament i realització, tant del codi com dels notebooks). Tot i que inicialment es van plantejar 8 setmanes pels dos mòduls primers i 4 pel darrer mòdul.

En cadascun d'aquests mòduls hi havia 1 persona liderant-lo, però l'altra persona estava donant suport en la seva realització. La Raquel tenia només 1 mòdul assignat perquè creiem que és el més extens i laboriós de tots 3 i, per tant, en el que majors dificultats a l'hora de desenvolupar-lo ens podem trobar. Tot i que durant la realització del treball s'ha vist que no és així, i que el més complex és el mòdul del laberint, per tant s'ha canviat de persona assignada per liderar el darrer mòdul.

A més, s'ha dedicat una setmana a la finalització de l'informe final, aquesta vegada, sí que dins del període desitjat inicialment.

Es va haver d'eliminar un període de temps de la planificació inicial.

4.1 Gantt de la programació de les tasques a realitzar

Es va plantejar la següent planificació:

T1. Resolució d'un laberint:

Persones que hi treballen: **Víctor**, Raquel
Durada: del 15 de març al 9 de maig.

T2. Reconeixedor de senyals, objectes i forats:

Persones que hi treballen: **Raquel**, Víctor
Durada: del 15 de març al 9 de maig.

T3. Unió dels dos mòduls:

Persones que hi treballen: **Víctor**, Raquel
Durada: del 10 de maig al 6 de juny.

T4. Setmana extra: Repassar incidències i errors

Persones que hi treballen: **Raquel**, Víctor
Durada: del 7 al 13 de juny.

T5. Preparar la presentació més informe final:
 Persones que hi treballen: **Raquel, Víctor**
 Durada: del 14 al 27 de juny.

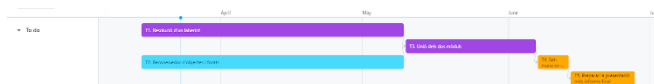


Figura 3. Gantt

Però durant la realització del treball es va veure que els mòduls 4 i 5 (períodes T2 i T1, respectivament) requerien més temps del previst. El mòdul 4 va estar finalitzat amb 2 setmanes d'endarreriment, mentre que el 5 ha requerit fins al dia 8 de juny. Això ens ha retrassat i hem hagut de fusionar el període T4 als altres mòduls, amb el risc de que es puguin colar errors degut a tenir un temps més apurat. També, degut a correcció d'errors i altres modificacions, el mòdul 4 ha requerit dedicar-li alguns dies més fins al dia 8 també, però com han sigut dies puntuals a la planificació final no està reflectit.

L'endarreriment dels mòduls 4 i 5 és, en major mesura, degut a que es va intentar fer servir TensorFlow en comptes de PyTorch i es van tenir problemes. Està detallat més endavant.

El mòdul 6 (període T3) s'ha vist desplaçat en el temps i es va dur a terme fins al dia 13 de juny.

També vam veure que l'informe final havia d'estar fet el dia 20 amb la seva versió més final possible, per tant durant el darrer període ha sigut per la presentació, que serà des del dia 20 fins a l'últim dia.

Per tant, la planificació queda, finalment, de la següent manera:

T1. Resolució d'un laberint:

Persones que hi treballen: **Víctor, Raquel**
 Durada: del 15 de març al 8 de juny.

T2. Reconeixedor de senyals, objectes i forats:

Persones que hi treballen: **Raquel, Víctor**
 Durada: del 15 de març al 23 de maig.

T3. Unió dels dos mòduls:

Persones que hi treballen: **Raquel, Víctor**
 Durada: del 23 de maig al 13 de juny.

T5. Preparar la presentació més informe final:

Persones que hi treballen: **Raquel, Víctor**
 Durada: del 14 al 27 de juny.

5 DESENVOLUPAMENT MÒDUL 5. LABERINT

En aquest apartat s'explica el desenvolupament del mòdul 5, que té per títol *Laberint*. S'expliquen les eines que s'han fet servir, la xarxa neuronal, les diferents classes

que del mòdul i les mecàniques de moviment i per recórrer el laberint.

5.1 Introducció

Per tal que les condicions siguin sempre les més semblants a l'entorn controlat on s'han fet les proves, pel terreny d'entrenament i de proves s'han dissenyat i fet un seguit de peces que es poden fabricar a cada centre de forma senzilla i assequible. Consisteix en fer uns quadrats de la mateixa mida de cartró i els camins són línies de cinta adhesiva de color fosc i acabat mate (no reflectant). Per tant, cada centre haurà de crear aquestes peces per poder treballar en aquest mòdul. El laberint està fet amb línies i no té parets físiques.

Un cop dissenyades i fetes les peces (1-2 hores), es va començar a realitzar el codi i l'entrenament de la xarxa neuronal que reconeix el camí de les peces.

Un cop entrenada, es van fer proves per veure que tot funciona correctament i llavors es va començar amb la part del moviment del robot segons les línies.

L'algorisme creat va guardant en una llista quin és el camí recorregut i quins s'han trobat però encara no s'han explorat (interseccions). Així, un cop arribat al final d'un camí torna enrere fent backtracking fins a una intersecció que encara li quedin camins no explorats i explora un nou camí.

Backtracking és un algorisme que permet prendre decisions i retornar a aquests punts un cop arribat al final de les accions d'aquella decisió presa. Un cop tornat al punt de la decisió, s'escull una opció no explorada, es torna a fer fins al final i així un altre cop fins a exhaurir les opcions de cada decisió.

Un cop aconseguit això, s'ha fet una funció que mostra per pantalla quin és el camí que s'ha trobat i el guarda en un .txt que es pot consultar un cop finalitzat el procés.

Finalment s'ha creat el notebook amb les explicacions oportunes i el codi del mòdul, traient les parts que els alumnes podran modificar i jugar amb elles per veure a quins resultats els porten.

5.2 TensorFlow i PyTorch

Es va intentar fer l'entrenament de la xarxa neuronal fent servir TensorFlow [3], ja que és la que fa servir Google, a més d'aprendre noves metodologies de les ja existent dins la plataforma als mòduls anteriors. Però sortien molts problemes i, per tant, es va cercar informació al respecte [4] però a cada cosa arreglada en surten molts altres errors. Es va consultar a professors de la universitat, però ells treballen amb PyTorch. Tot i que sí van intentar ajudar, no es va arribar a trobar una solució al problema.

Després del temps invertit, es va contemplar la idea de canviar a PyTorch, no només perquè si tornés a sortir

6 EXPERIMENTS MÒDUL 5. LABERINT

En aquest apartat es mostren els experiments realitzats per a que la tasca dels i de les alumnes sigui més còmoda i tinguin una base de la qual partir per arribar a les conclusions desitjades.

6.1 Peces de camí

Per poder fer les fotos de les diferents opcions de camí que es trobarà el robot s'ha de construir un laberint o simular cada cosa diferent i fer les respectives fotos. Per fer la tasca més fàcil, i evitar així una despesa de cinta adhesiva excessiva, s'ha optat per fer unes peces de cartró amb les opcions de camí, així no només serà més fàcil la tasca de construir el laberint i treure'l, sino que a més es poden fer moltes combinacions amb el mateix material. Es pot veure la plantilla imprimible de les peces a la figura 5, que estarà en format pdf a la plataforma, i un exemple d'una peça de camí acabada a la figura 6.

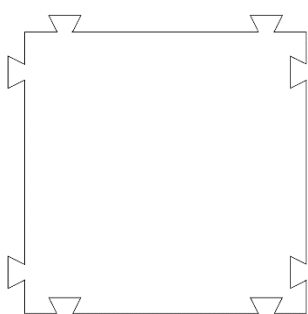


Figura 5. Model de peça



Figura 6. Peça feta

Les peces tenen dents per poder-se connectar entre elles, així si durant el procés del seguiment del laberint alguna peça es mou, tot el laberint es mou, evitant que el nano JetBot es perdi o faci alguna cosa rara degut a quelcom extern.

6.2 Mostra d'imatges

Per tal d'evitar que els alumnes hagin de fer milers de fotos, s'ha buscat un número petit d'imatges a fer per cada classe. Es va començar amb 10 fotos per classe, però la xarxa fallava molt i es va canviar a 20, notant una clara evolució positiva.

En les proves efectuades, es classificaven correctament les peces quan es mostrava d'una en una, però a l'hora de posar-lo en un laberint no classificava correctament, tot i tenir un accuracy en validació d'un 86% amb 20 imatges per classe per efectuar l'entrenament.

Vist això, s'han anat afegint imatges a cada classe fins a trobar que 50 és el nombre mínim d'imatges que s'han de fer per cada classe.

S'ha creat un dataset de validació d'unes 10 imatges per corroborar que tot funcioni correctament. Ara, amb les 50 imatges de test i 10 de validació per classe, la xarxa falla molt poc, amb un accuracy en test d'un 95% i un 90% en validació.

7 RESULTATS MÒDUL 5. LABERINT

Per veure els resultats d'una manera clara s'han fet una gràfica de convergència amb l'accuracy de la validació i del test, i una matriu de confusió on es veu quines són les sortides que hauria de donar la xarxa i quines són les resultants.

7.1 Amb 20 imatges per classe

Primer s'analitzarà què passava amb 20 imatges.

A la figura 7 es pot veure la gràfica de convergència on es mostra l'accuracy a cada epoch de les proves de test i de validació.

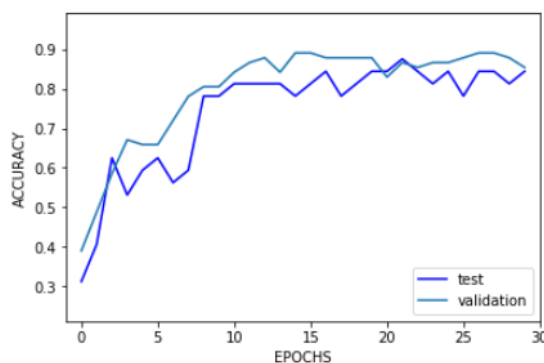


Figura 7. Gràfica de convergència 20 img

Són molt similars les 2 gràfiques, tenen la mateixa forma i els valors no disten gaire els uns dels altres. L'algorisme guarda com a millor model el que dona un accuracy en test més alt, essent l'obtingut en l'epoch 21 el més alt, amb un accuracy en test del 87%. L'accuracy de validació en l'epoch 21 és del 86%.

A la figura 8 hi ha la matriu de confusió de la sortida de la xarxa. Es pot veure el percentatge d'encerts de la xarxa en les classes de les columnes que coincideixen amb les classes de les files (diagonal).

	Creu	Dreta	Esqu.	EsDr.	Final	Recte	ReDr.	ReEs.
Creu	64%	0%	5%	0%	0%	0%	23%	5%
Dreta	0%	84%	7%	0%	0%	0%	7%	0%
Esqu.	0%	0%	100%	0%	0%	0%	0%	0%
EsDr.	0%	0%	0%	92%	7%	0%	0%	0%
Final	0%	0%	0%	0%	100%	0%	0%	0%
Recte	8%	0%	8%	0%	0%	66%	8%	8%
ReDr.	0%	0%	0%	0%	0%	0%	100%	0%
ReEs.	0%	0%	6%	0%	0%	0%	6%	86%

Figura 8. Matriu de confusió 20 img

Veient la matriu, es pot observar que algunes classes tenen un encert molt alt. Les classes Creu i Recte són les que tenen un percentatge d'encert més baix, només 2 de cada 3 imatges es classifica correctament, mentre que, en el cas de la classe Creu, 1 de cada 4 imatges es classificarà com un intersecció Rectedreta. Es va considerar que seria millor fer mostres de més imatges per cada classe.

7.2 Amb 50 imatges per classe

Aquí s'analitza què passa ara amb 50 imatges.

A la figura 9 es pot veure la gràfica de convergència, com a l'apartat anterior, però ara amb 50 imatges.

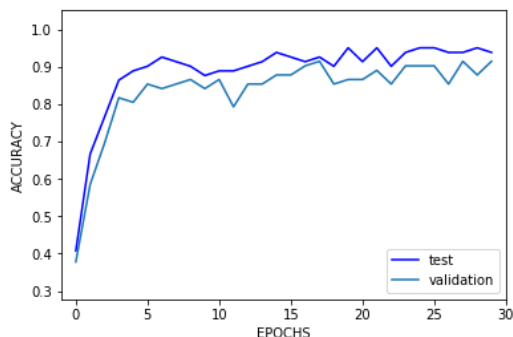


Figura 9. Gràfica de convergència 50 img

Es torna a veure que hi ha similituds entre les 2, però aquest cop sí que s'aprecien diferències. Tot i així, l'algorisme guarda els valors que han donat un accuracy en test més alt. En aquest cas, seria, també, a l'epoch 21, amb un 95 % d'accuracy de test i un 90% de validació.

A la figura 10 es pot veure la matriu de confusió, però ara a la que correspon al cas de la mostra de 50 imatges per cada classe.

	Creu	Dreta	Esqu.	EsDr.	Final	Recte	ReDr.	ReEs.
Creu	85%	0%	0%	0%	5%	0%	5%	5%
Dreta	0%	90%	5%	0%	0%	0%	0%	5%
Esqu.	0%	0%	96%	0%	0%	0%	0%	4%
EsDr.	0%	0%	5%	90%	5%	0%	0%	0%
Final	0%	0%	11%	0%	88%	0%	0%	0%
Recte	0%	0%	0%	0%	4%	86%	4%	4%
ReDr.	0%	0%	0%	0%	0%	5%	94%	0%
ReEs.	0%	0%	4%	0%	0%	0%	0%	95%

Figura 10. Matriu de confusió 50 img

Observant aquesta nova taula, alguns valors han empitjorat lleugerament, però altres han millorat de forma considerable. El percentatge més baix d'encert està en un 85 %, mentre que abans el més baix era del 66 %, ambdós en el cas de la classe Creu.

S'ha decidit quedar-se amb aquests nous valors, tot i que no són del tot alts. Això és degut perquè s'està buscant un nombre mínim d'imatges a fer per cada classe i amb 50 ja es pot treballar de forma acceptable. Tot i que als i les alumnes se'ls indica que 50 és el mínim a realitzar i se'ls convida a fer-ne moltes més. A més, pot ser una bona manera de que experimentin ells mateixos quines són les diferències entre entrenar amb més i amb menys imatges.

Per entrenar la mostra de 50 imatges de test més les 10 de validació s'han requerit 531,54 segons, el que és el mateix 8 minuts, 52 segons i 54 mil·lèsimes.

8 NOTEBOOK MÒDUL 5. LABERINT

La plataforma educativa està fet en un Jupyter Notebook, per tant s'han de fer totes les activitats que els i les alumnes podran fer dins del mateix Jupyter.

El mòdul s'ha separat en 6 apartats:

0. Introducció
1. Recollida d'imatges
2. Entrenament
3. Reconeixement i moviment
4. Guardar el camí
5. Backtracking

8.0 Introducció

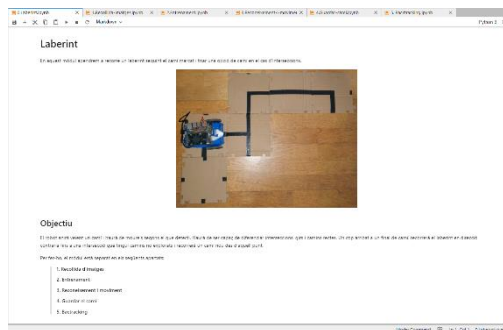


Figura 11. Plataforma. Apartat 0

En aquest apartat és on es defineix quin és l'objectiu del mòdul i tots els apartats que tindrà. S'introdueixen les peces i es demana que es fabriquin.

Trobem, també, una secció que indica quins són els coneixements previs requerits per poder dur a terme el mòdul. Hi ha blocs de codi que poden executar i provar, a part d'una breu explicació de cadascun d'aquests coneixements.

8.1 Recollida d'imatges

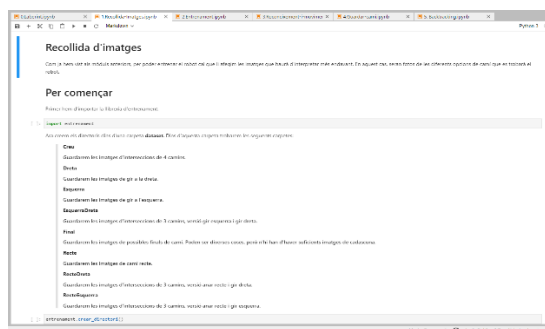


Figura 12. Plataforma. Apartat 1

En aquest primer apartat s'indica com es recolliran les imatges. S'explica que es farà servir la càmera del robot per fer-les i es guardaran en un directori anomenat dataset en el que hi trobarem 8 carpetes, una per cada classe.

Hi ha un sistema de botons que prendran les fotos i les guardaran a la carpeta corresponent segons la seva classe. Al costat de cada botó hi ha un comptador d'imatges, així es pot saber en tot moment quantes imatges hi ha en cada classe sense haver de comptar una per una les imatges de cada carpeta. A sobre dels botons podem veure el que

està detectant la càmera. Tot això ho podem veure a la figura 13.



Figura 13. Botons

8.2 Entrenament

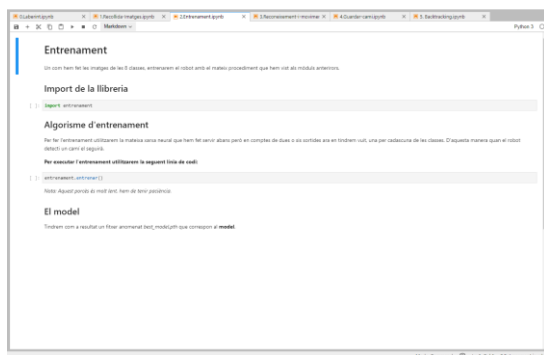


Figura 14. Plataforma. Apartat 2

En aquest apartat s'indica com efectuar l'entrenament del robot, tot i que ve a ser el mateix que per als mòduls anteriors, per tant serà un apartat ràpid de fer i sense complicacions. Tot i ser ràpid de fer, no és ràpid d'acabar, ja que l'entrenament pot trigar molta estona, depenent de la quantitat d'imatges que s'hagin fet.

8.3 Reconeixement i moviment

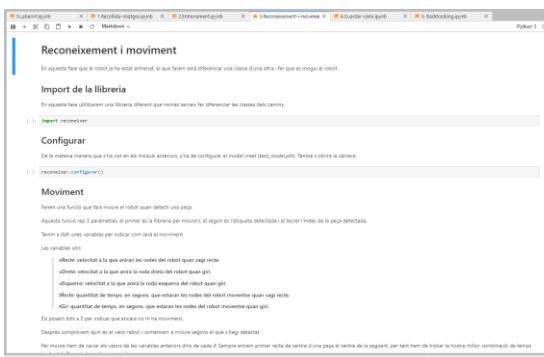


Figura 15. Plataforma. Apartat 3

Aquí es detalla com fer la predicció a partir del que està veient la càmera, com poder triar una de les classes i com fer el moviment del robot segons el criteri de selecció donat.

Els i les alumnes tindran l'oportunitat de triar com serà la funció que farà les decisions i escollint el criteri de selecció. Se'ls hi donen funcions d'exemple amb una explicació, però són lliures de modificar-les al seu gust i fer les

proves pertinents. D'aquesta manera poden aprendre tant dels seus mèrits com dels seus errors, experimentant ells mateixos els resultats.

8.4 Guardar el camí

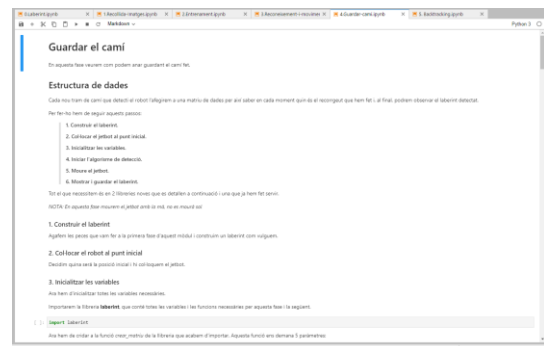


Figura 16. Plataforma. Apartat 4

En aquest apartat es detalla com va guardant el camí recorregut el robot.

Aquest camí guardat és el que farà servir el robot després per poder tornar enrere fins a una intersecció que li quedin camins sense explorar mitjançant Backtracking, i així fins a quedar-se sense interseccions lliures.

Per poder veure com funciona, primer es demana fer una funció de decisió per trobar la classe a la que pertany la peça de camí que té davant. Es pot fer servir la de l'apartat anterior, però aquest cop es treurà la part del moviment.

Un cop creada aquesta funció, es demana anar col·locant el Jetbot davant de la següent peça que tocaria per així es pugui anar creant el laberint.

Un cop finalitzat, s'explica com poder mostrar el laberint i com es pot guardar en un arxiu de text. A la figura 17 podem veure un exemple de la mostra del laberint. Les línies indiquen el camí del laberint i les x representen llocs no disponibles per moure's.

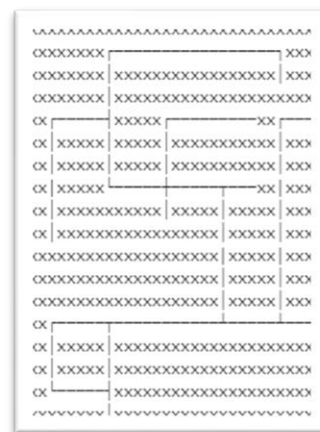


Figura 17. Exemple de laberint

8.5 Backtracking

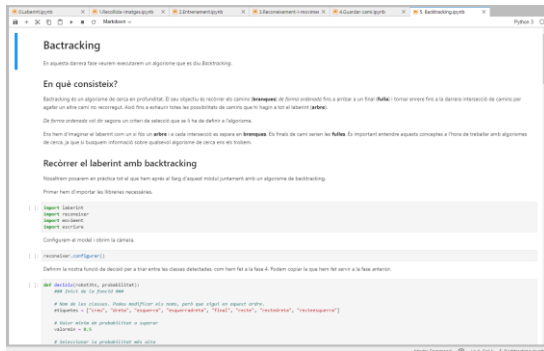


Figura 18. Plataforma. Apartat 5

Per últim, en aquest apartat el robot ja serà capaç de fer tots els objectius del mòdul i els i les alumnes podran experimentar-ho.

S'explica en què consisteix el Backtracking de forma molt senzilla i com aplicar-lo al nostre mòdul. Per tant, es dona a entendre que el que han fet a l'apartat anterior ja era un algorisme de Backtracking, però que el moviment i la presa de decisió s'han fet de forma manual, movent amb la mà o indicant les noves coordenades i sentit.

Per a la funció de Backtracking es dona un algorisme ja definit amb indicacions de com funciona mitjançant comentaris dins de la pròpia funció.

Els i les alumnes han de fer la seva pròpia funció de moviment i la de decisió, tot i que en aquesta darrera poden agafar la que ja van fer a l'apartat 3 del mòdul i fer-li una petita modificació (indicada a la plataforma). Tot i així, hi ha definides una funció de cada a la plataforma per si el professor veu que els i les alumnes encara no estan preparats per fer la seva pròpia funció.

Un cop vist que tot funciona correctament, es demana als i les alumnes que facin el seu propi algorisme de backtracking empleant els nous coneixements adquirits. Tot i que això últim només si el professor veu que tenen el nivell suficient, és una part opcional. L'objectiu d'aquest apartat és que entenguin com funciona, però que siguin capaços de fer un Backtracking.

9 DESENVOLUPAMENT MÒDUL 6. LABERINT AMB OBJECTES

El mòdul 6 és la unió dels mòduls 4 i 5. El mòdul 4 està detallat al TFG de la Raquel Navarro, d'enguany també. El mòdul 5 és el que s'ha explicat i analitzat en el present treball.

A mesura que s'han anat completant els 2 mòduls anteriors s'han anat afegint en aquest. S'ha fet servir el mateix funcionament i material que a cada mòdul: ús de peces amb els camins, senyals, etc..., Però amb algunes

modificacions oportunes, com per exemple combinar algunes classes d'ambdós mòduls.

Els aspectes modificats respecte el mòdul 4 es detallen a l'altre TFG. En aquest apartat veurem els canvis respecte el mòdul 5.

Al mòdul anterior es recorria el laberint amb l'únic objectiu de poder recórrer i dibuixar tot el laberint. En aquest té l'objectiu afegit de buscar objectes i retornar un recorregut. A més de tenir unes restriccions extres, el senyals i els forats.

Cada objecte dona una sèrie de punts, definits pels i les alumnes, per fer que el camí retornat sigui en el que hi hagi l'objecte amb més punts. També comprova que si s'ha trobat el mateix objecte, comprova que el recorregut sigui més curt i, si és el cas, retorna aquest recorregut.

10 NOTEBOOK MÒDUL 6. LABERINT AMB OBJECTES

Aquest mòdul s'ha separat en 5 apartats:

0. Introducció
1. Recuperar el dataset
2. Recollida d'imatges
3. Fer-lo aprendre
4. Recorrer el laberint

10.0 Introducció

En aquest apartat és on es defineix quin és l'objectiu del mòdul i tots els apartats que tindrà.

10.1 Recuperar el dataset

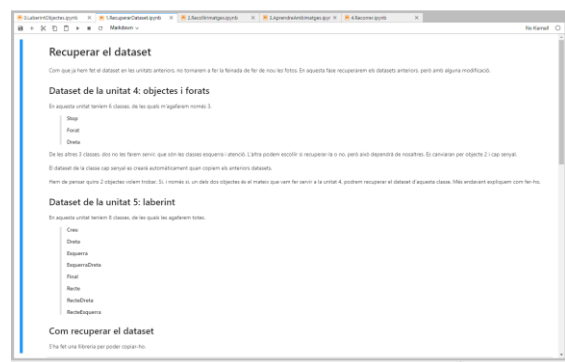


Figura 19. Plataforma. Apartat 1, mòdul 6

En aquest apartat es recupera tot el data set fet en els dos mòduls anteriors. Es fa així per evitar que es facin les imatges de nou, ja que en són moltes. S'agafen totes les imatges de la unitat 5, però només s'agafen les imatges de 3 classes de la unitat 4, degut a que es canvien respecte aquesta unitat.

Aquestes imatges se separen en 2 datasets diferents per tractar-los de distintes maneres. Algunes fotos de la unitat

5 es copien a una classe nova que s'ha de crear per al dataset de les senyals, la classe que indica que no hi ha cap senyal.

10.2 Recollida d'imatges

Aquí és on creen dues noves classes que calen per al dataset de les senyals. Aquest cop només hi ha 2 botons, un per cada classe. Funciona igual que per als altres mòduls.

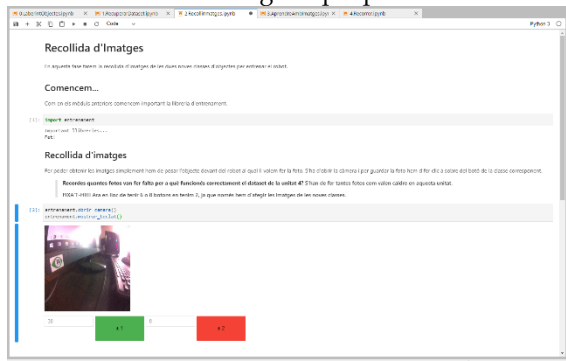


Figura 20. Plataforma, Apartat 2, mòdul 6

10.3 Fer-lo aprendre

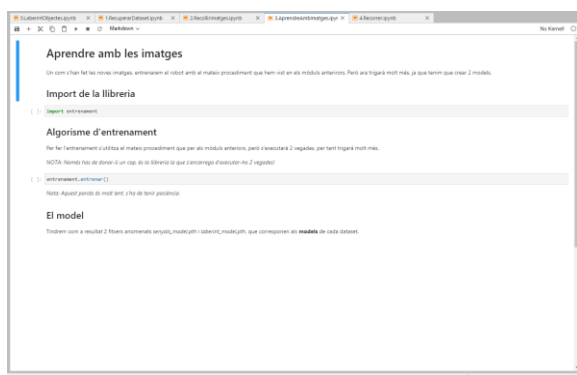


Figura 21. Plataforma, Apartat 3, mòdul 6

Es fa de la mateixa manera que per als altres mòduls, però amb la peculiaritat que ara s'obtenen 2 models diferents, un per cada data set.

10.4 Recorrer el laberint

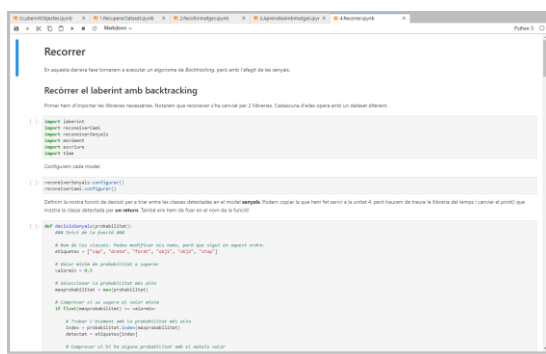


Figura 22. Plataforma, Apartat 4, mòdul 6

En aquest apartat els i les alumnes tornen a tenir la oportunitat de fer servir un algorisme de backtracking i al final poden fer el seu propi.

Aquesta vegada s'ha de tenir present que s'han de detectar i processar les senyals, els objectes i els forats, a més del camí com fèiem al punt 8.5. També, els i les alumnes han de notar que s'han de crear unes variables noves per guardar el valor més alt de punts trobat i quin és el recorregut més curt que ens dona aquest valor.

11 CONCLUSIÓ

En aquest TFG s'ha treballat en una plataforma educativa per apropar la intel·ligència artificial i les ciències de la computació a estudiants dels últims cursos de la ESO amb nano jetbots. Concretament, s'ha desenvolupat un paquet consistent en resoldre un laberint de forma automàtica, separat en dos mòduls. En el primer mòdul els estudiants han de fer que el robot reconegui les diferents peces que formen el camí i aprenen experimentant, de forma manual, l'algorisme de backtracking. En el segon mòdul, desenvolupat conjuntament amb la Raquel Navarro, els estudiants han de fer que el robot recorri el laberint però ara té restriccions de pas i gir i ha de trobar uns objectes que li donen un seguit de punts.

En aquest TFG s'han preparat els mòduls per tal que els estudiants puguin explorar els diferents algorismes sense haver-se de preocupar per la programació i els detalls, tant de la xarxa neuronal utilitzada com de l'algorisme de backtracking. A més, s'ha analitzat l'eficiència de la xarxa perquè els estudiants puguin veure el seu comportament i es puguin plantejar diferents preguntes.

S'han complert tots els objectius previstos inicialment de forma satisfactòria.

Com a conclusió a nivell més personal, s'ha vist és que no és el mateix fer un algorisme per un mateix que per terceres persones, sobretot si és per a ensenyar a alumnes. Això ja s'ha vist durant el transcurs del grau, però mai s'ha fet un algorisme per ensenyar a terceres persones i aquí és un dels llocs on s'ha hagut de parar una mica i pensar quines parts poden editar i quines han de ser més ocultes.

També, com a observació, s'ha vist com és d'important tenir temps extres pels contratemps que puguin sorgir. Tot i que pel TFG ja es tenia un temps una mica ajustat per tot el que es volia fer i ha esdevingut en poc temps per actuar.

AGRAÏMENTS

Primer de tot, vull agrair a la tutora que hem tingut la Raquel i jo, ja que ens ha hagut d'aguantar fins al final i ens ha ajudat a resoldre alguns problemes o ens ha donat

alguna idea per evitar que ens atasquessim.

A en Felipe Lumbreras, que vam haver de demanar-li ajuda amb el tema de TensorFlow i va dedicar una bona estona a ser amb nosaltres.

A la Raquel, ja que ens hem estat donant suport mutuament i idees de millora en el TFG durant tot el transcurs del mateix.

Finalment, vull agrair a l'Associació Juvenil TriCric ^[6] degut a que ens ha cedit el seu domini per a poder unir els dos treballs en una sola plataforma sense la necessitat d'estar la Raquel i jo junts presencialment, així com poder treballar amb els jetbots a distància.

BIBLIOGRAFIA

- [1] Alan Roldán Maillo – Follow the leader amb nano jetbots, plataforma educativa – Escola d'Enginyeria Febrer 2020.
- [2] Spike Prime:
<https://education.lego.com/es-es/start/spike-prime#Introducción>
- [3] Implementació AlexNET amb TensorFlow:
<https://towardsdatascience.com/implementing-alexnet-cnn-architecture-using-tensorflow-2-0-and-keras-2113e090ad98>
- [4] Solució problema dataset, ragged:
<https://stackoverflow.com/questions/47580716/how-to-input-a-list-of-lists-with-different-sizes-in-tf-data-dataset>
- [5] Error Triggered amb PyTorch:
<https://towardsdatascience.com/cuda-error-device-side-assert-triggered-c6ae1c8fa4c3>
- [6] Associació Juvenil TriCric: <http://www.tricric.cat/>
(Podria ser que estigui apagat el servidor).